Utilizing Expert Knowledge to Support Medical Emergency Call Handling

Carsten Maletzki[1,*], Eric Rietzke[1,2] and Ralph Bergmann[1,3]

[1]German Research Center for Artificial Intelligence (DFKI)
Branch University of Trier, Behringstraße 21, 54296 Trier, Germany
[2]LiveReader GmbH, Zur Imweiler Wies 3, 66649 Oberthal, Germany
[3]University of Trier, Behringstraße 21, 54296 Trier, Germany

Abstract

Medical emergency calls require fast decisions from call takers about triage and appropriate responses. Call takers approach this challenge by deriving decisions from mental pictures they create by assessing available information with their expert knowledge. Established questionnaire-based support systems in this context experience hesitant acceptance while an alternative that is currently researched suffers from its complex approach to utilizing formalized expert knowledge. This paper addresses the latter by designing an Ontology- and Data-Driven Expert System (ODD-ES) for call takers of medical emergency calls. ODD-ES aims at supporting call takers with recommendations regarding decisions and questions that result from inferred artificial mental pictures. The knowledge base used to infer artificial mental pictures builds on semantically modeled functions to achieve maintainability and an integration of symbolic and subsymbolic Artificial Intelligence (AI). To make recommendations and handle responses of call takers, ODD-ES proposes a component called Copilot that will be in the focus of our future work.

Keywords

Expert System, Ontology- and Data-Driven Process Support, Medical Emergency Calls

1. Introduction

Disruptive events like pandemics or natural disasters demand a broad range of mitigating measures to achieve resilient societies and economies. Some of these measures are defined by call takers of medical emergency calls who often perform patient triage under time pressure and decide about the deployment of emergency resources like ambulances. To navigate this difficult area, call takers ground their decisions on mental pictures they create by applying their expert knowledge to information obtained during the call [1].

Established systems to support medical emergency calls partially substitute the need for mental pictures of call takers by prescribing decision tree like questionnaires that vary in their strictness regarding order and scope of questions. Although these systems are deemed beneficial to the quality of emergency call handling, strict systems are criticized for their lack of flexibility

8th Workshop on Formal and Cognitive Reasoning, September 19, 2022, Trier, Germany
*Corresponding author.
Carsten.Maletzki@dfki.de (C. Maletzki); Eric.Rietzke@dfki.de (E. Rietzke); Ralph.Bergmann@dfki.de (R. Bergmann)
0000-0003-0983-5015 (C. Maletzki); 0000-0001-5252-4859 (E. Rietzke); 0000-0002-5515-7158 (R. Bergmann)

while loose approaches can lead to forgotten questions that put patients at risk [2, 3]. In contrast, an alternative approach that is currently being researched aims at adaptive questionnaires and decision support by utilizing rule-based expert knowledge to assess available information [4]. Although a first evaluation of the underlying Ontology- and Data-Driven Business Process Model (ODD-BP) has been promising, rule-based formalization of expert knowledge turned out to be impractical at scale [5].

We address this issue by proposing an Ontology- and Data-Driven Expert System (ODD-ES) for call takers of medical emergency calls that relies on an approach to formalize expert knowledge that is tailor-made for ODD-BP. This approach was iteratively developed on the basis of knowledge acquisition workshops we conducted together with experts from the German emergency medical services. As expert systems try to mimic the thinking, skill and intuition of experts [6], ODD-ES derives recommendations for decisions and questions from inferences added to a knowledge base that result from applying formalized expert knowledge to available emergency information. The knowledge base thereby is designed to integrate symbolic and subsymbolic approaches to Artificial Intelligence (AI) while a component called Copilot consults the call taker to communicate recommendations and handle responses. Therefore, this paper contributes to the research demand towards human-AI interaction in medical emergency call handling [7].

The following sections start by laying out relevant foundations and related work. As a basis for the design of ODD-ES, we will analyze the processes of medical emergency calls and explain how they are represented in ODD-BP. Afterwards, we will introduce ODD-ES, discuss each of its components, sketch future work and conclude our findings.

2. Foundations and Related Work

Over the last decades, advancements in medical emergency call support were often driven by outstanding or deficient call taker performances which should either be repeated or avoided [8]. The resulting support approaches of today can be divided by the degree to which they prescribe the order of call taker tasks and questions while acting either more like loose guidelines or strict protocols [2]. Although there is a consensus that these system are beneficial to the quality of emergency call handling, they suffer from hesitant acceptance in Germany [2, 3, 9]. To the best of our knowledge, apart from our work, only a single approach is currently being researched that is designed to work on top of established support systems. This approach uses neural-networks to identify based on transcribed caller statements whether the patient has an out-of-hospital cardiac arrest, which performs slightly better than call takers [10].

Medical emergency calls belong to the category of so-called Knowledge-intensive Processes (KiPs) [11]. KiPs are characterized by their strong focus on data and information, while their execution depends on the decisions made by process participants utilizing their knowledge [12]. The problem of providing ideal support for KiPs is an open research topic – however, there are indications that data-centric business process modeling poses a promising direction [11]. In the context of KiPs, several data-centric approaches to business process modeling have been developed so far [4, 12, 13]. Compared to traditional process models that focus on a specific order of tasks (called control-flow), data-centric approaches focus on the data that is required for task execution. This paradigm shift results in a process logic that is driven by available data rather than insisting on a pre-defined sequences of tasks.

Support for KiPs should not only regard their data-centric nature but also help process participants utilize their knowledge. Both could possibly be addressed by integrating data-driven process technology with expert systems. Expert systems evaluate case data by applying formalized expert knowledge to mimic the thinking, skill and intuition of experts while usually containing an inference engine, a knowledge base and a task-specific database [6]. In combination with data-driven process support, an expert system would utilize expert knowledge to evaluate process data to identify possible decisions and recommend them to process participants. The area of expert systems has thereby been a subject of research for decades while they have also been implemented on the basis of ontologies [14]. To the best of our knowledge, only our own research is currently aiming at an integration of an ontology- and data-driven process system with an expert system to support KiPs [5].

3. Analysis of Medical Emergency Calls

Whenever citizens in Germany face medical or firefighting-related emergencies, they can call the emergency number 112 to request professional help. Their calls are handled by call takers in emergency control centers who assess their emergencies and decide about appropriate responses. When handling medical emergency calls, call takers perform triage and decide about the type of required emergency resources. The actual deployment of emergency resources is subsequently handled either by the call takers themselves or dedicated dispatchers, whereby the exact competence depends on organizational factors and operational circumstances.

Møller et al. [1] recently introduced a conceptual model that summarizes their findings about the call takers' perception of medical emergency call handling. This model, which is shown in figure 1, describes medical emergency calls as processes whose executions are strongly influenced by the caller's and call taker's context. These influences materialize in the caller's and call taker's view and thus influence their behavior. The caller is in that sense influenced by his/her motive for calling, situation and ability to percept and verbally present the problem to the call taker (figure 1: Caller Influences). Situational influences can thereby arise for example from harmful events like accidents, demographic factors and from a possible professional medical

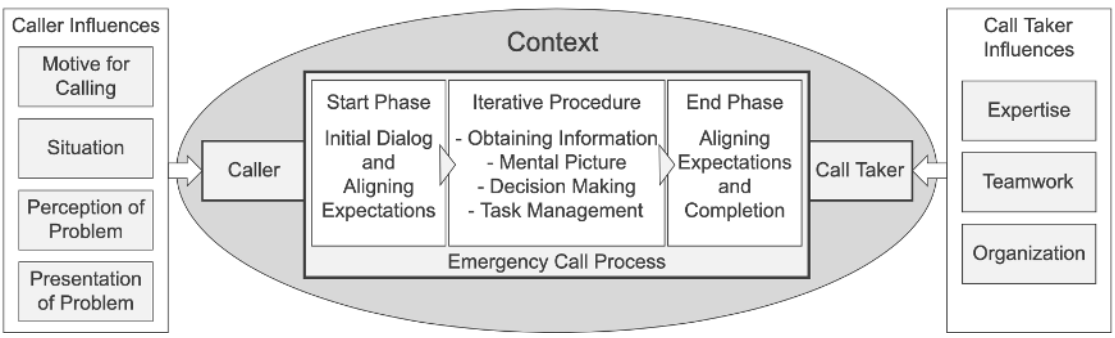

Figure 1: Model of Medical Emergency Call Handling based on Møller et al. [1]

background. Furthermore, the ability or willingness to assess the patient can influence the caller's behavior during the call and therefore restricts the ability of the call taker to handle the call appropriately. Influences on the call taker originate for example from his/her ability to apply and exchange knowledge and information with colleagues and from organizational factors like the characteristics of the tools used to support medical emergency call handling (figure 1: Call Taker Influences). The emergency call process itself has an iterative procedure at its core that is framed by a start and end phase required to perform an alignment of expectations between caller and call taker (figure 1: Emergency Call Process). When performing the iterative procedure, the call taker has to obtain relevant information from the caller by asking the right questions in order to get a clear mental picture about the reported emergency. The mental picture thereby is created by applying expert knowledge to interpret the information given throughout the emergency call. Based on the resulting mental picture, the call taker has to decide about the patient's condition while possibly determining a suspected diagnosis. Afterwards, the call taker decides which type of emergency resource would be appropriate to handle the case and manages subsequent tasks like handing the case over to the dispatcher.

4. Representing Emergency Calls with ODD-BP

The Ontology- and Data-Driven Business Process Model (ODD-BP) that is extended in this work, proposes a metamodel for data-centric process models that enables the realization of a data-driven process system based on ontologies. ODD-BP has already been introduced in detail and implemented in a base ontology [4]. To the scope of this paper, we focus on the most relevant concepts shown in figure 2 and their application required to design ODD-ES. A process model in ODD-BP is represented by an individual that either instantiates the class of *process definitions* or *process instances* and *contains* individuals of process elements. While *process definitions* are templates for processes, *process instances* represent single enactments. Since ODD-BP follows a data-centric approach to process modeling, the input and output relations between the process elements *tasks*, *dataobjects* and *attributes* (input: *required_by*; output: *delivers*) lie at the core of

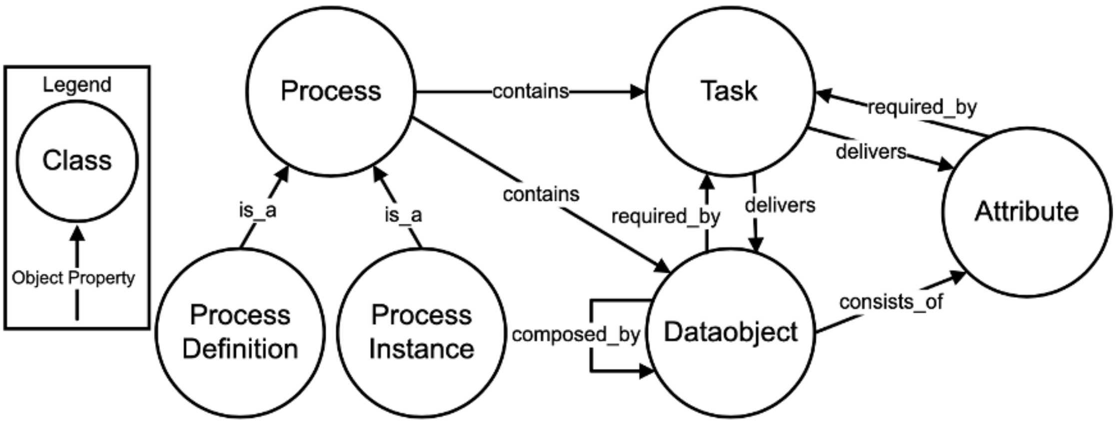

Figure 2: Excerpt of the ODD-BP Metamodel [4]

its conceptualization. While *tasks* represent units of work inside a *process*, *dataobjects* represent entities whose *attributes* are processed during *task* execution. To enhance the manageability of larger *dataobjects* they can be divided into composing dataobjects that cluster thematically related *attributes*. The actual data value of an *attribute* is stored as a literal via a datatype property on the corresponding *attribute* individual. To specify the exact meaning of these values in the context of an application scenario, *attribute* and *dataobject* individuals further instantiate domain-specific classes describing for example that an entity is a person (*dataobject*) that has a name (*attribute*). When implemented in a process system, ODD-BP further aims at supporting the execution of *process instances* and therefore contains knowledge about the executability and relevance of *tasks*. This knowledge is used by an inference engine to identify the executability of *tasks* and the degree to which a *task* execution is relevant to achieve process goals [4].

To apply ODD-BP to medical emergency calls, we modeled entities whose data influences the process execution as *dataobjects* and *attributes* in a *process definition*. The resulting data model was thereby developed on the basis of the conceptual process model from Møller et al. [1] and in collaboration with domain experts. The main *dataobjects* of this data model are the ones of the patient and the caller. As these *dataobjects* can get complex, they have been divided into composing *dataobjects* representing single aspects like the problem that is reported. This *dataobject* of the problem includes, for example, *attributes* that describe symptoms and diagnoses that are recorded or determined throughout an emergency call. Since *dataobjects* represent all relevant entities that are involved in the process, it is possible that multiple patients can occur in a single *process instance*. Questions that could be relevant to ask by the call taker are further represented as *tasks* linked to the *dataobjects* and *attributes* they address.

5. Ontology- and Data-Driven Expert System

This section introduces ODD-ES – an expert system that extends ODD-BP to support call takers in medical emergency calls. The following subsections start with a general overview of the operating principles of ODD-ES in the context of ODD-BP and afterwards explain each of its components in detail.

5.1. Operating Principles of ODD-ES in Context of ODD-BP

Implemented in a process system, ODD-BP aims at contributing to medical emergency calls by recommending tasks for execution in which call takers would, for example, have to ask questions that obtain relevant information from the caller. In this context, ODD-ES provides the foundation for ODD-BP as it identifies information that is relevant enough to justify the time that it takes to ask questions about them. ODD-ES approaches this in a first step by utilizing formalized expert knowledge to generate inferences that are added to the knowledge base of the system. In this context, the sum of all inferences resemble the so-called artificial mental pictures of the system. Afterwards, ODD-ES determines the missing information that would contribute the most to a clarification of the artificial mental picture in order to make appropriate decisions. In case that available information suffices for a decision already, ODD-ES detects and recommends this to the call taker who is free to either adopt or reject it. The same is done when ODD-ES identifies gaps in the artificial mental picture that should be clarified by obtaining further information. Whenever the call taker accepts a proposal, ODD-ES modifies the process instance accordingly to influence which tasks, i.e. questions, are further proposed by ODD-BP.

Figure 3 illustrates how the components of ODD-BP and ODD-ES interact with each other to achieve the described behavior. Initially, the emergency information available to a process instance of ODD-BP is the input of the knowledge base of ODD-ES. This knowledge base contains formalized expert knowledge that is applied to the emergency information by an inference engine that infers an artificial mental picture. Subsequently, the resulting state of the knowledge base is analyzed to identify gaps in the artificial mental picture that should be clarified or decisions that can be made already based on the available information. This analysis is performed by a component of ODD-ES called Copilot. The name Copilot originates from its role of being an assistant who consults the call taker to recommend decisions and possible directions to obtain information. If the call taker accepts a recommendation, the Copilot has to modify the process instance of ODD-BP to reflect the impact this has on the process. This can then lead to a change in recommended tasks and questions as a result of the next application of the inference engine in ODD-BP.

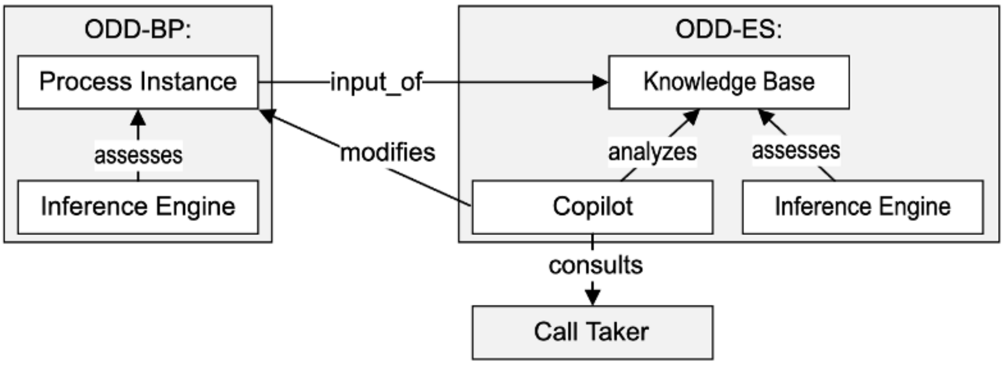

Figure 3: Operating Principles of ODD-ES

5.2. Knowledge Base & Inference Engine

In the following, the structure of the knowledge base of ODD-ES is developed on the basis of the experiences made in the context of ODD-BP. So far, ODD-BP has used the Semantic Web Rule Language (SWRL)[1] to express and apply expert knowledge in medical emergency calls [5]. Although SWRL is easy-to-use, the resulting rules tend to be complex in the context of ODD-BP. This complexity arises from the situation that rules have to reflect extensive structures of the process instance to work as intended. As a result, this approach leads to substantial maintenance effort and since it cannot be reduced by simply switching to another already existing language, we subsequently develop an alternative approach.

The knowledge base of ODD-ES aims at a clear separation towards the process instance at design time while only reflecting minimal structures of the process instance, which is generally seen as advantageous with regard to the maintainability of the expert system [6]. The most

[1]SWRL Specification: https://www.w3.org/Submission/SWRL/

important building blocks of the knowledge base in ODD-ES are semantically modeled functions. The conceptualization of these functions is similar to OWL-S, a semantic markup for web services[2], but our approach is significantly less complex.

Figure 4 depicts the structure of semantically modeled functions in ODD-ES. Functions in ODD-ES have at least one input and one output parameter while the type of each individual is expressed by an instantiation of a domain-specific ontology class. Thus, using an example from medical emergency calls, a function can be of the type 'Fever Threshold' taking the 'Body Temperature' as an input to return whether someone has a 'Fever' as an output. The

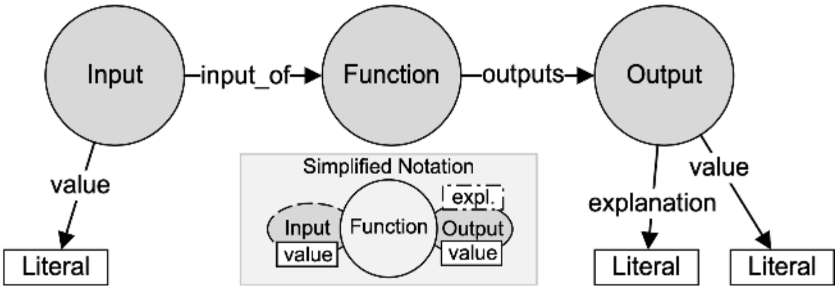

Figure 4: Semantically modeled Function in ODD-ES

execution of such functions by an inference engine is carried out on the basis of the data values available on the respective input individuals. This implies that before any inferencing can take place, attribute values from process instances in ODD-BP have to be made available to appropriate input individuals. This step is performed by the inference engine of ODD-ES which links attributes from a process instance to input individuals of ODD-ES if they instantiate the same domain-specific classes. Output individuals are also regarded in this step as it allows returning inferences back to the process instance. This is for example required to handle decisions that ODD-ES proposed to the call taker that were accepted. Establishing these links based on domain-specific ontology classes circumvents the issue of having to describe extensive structures of the process instance. However, this assumption requires uniqueness to produce correct inferences. We will discuss this issue later in detail and introduce a concept that should suffice the requirements of our application scenario.

A strength of utilizing functions for reasoning is that they can be implemented in any way. This allows combining simple logical operations with complex neural networks in a homogeneous knowledge base. As a result, the inference engine takes over the role of a runtime environment that executes program code that was registered, for example via annotations of ontology classes as it is done in the programming library OWLready2[3]. Utilizing functions for inferencing has already been introduced by the advanced features of the Shape Constraint Language (SHACL)[4]. However, SHACL does not perform function execution based on data values of linked input parameters and further does not provide output individuals. Both is done by ODD-ES to facilitate maintenance, the inferencing procedure and the analysis performed

[2]OWL-S W3C Submission: https://www.w3.org/Submission/OWL-S/
[3]OWLready2 Documentation: https://owlready2.readthedocs.io/en/latest/intro.html
[4]SHACL Advances Features Specification: https://w3c.github.io/shacl/shacl-af/

by the Copilot component as it will be discussed in the rest of this section. Output individuals are further used in ODD-ES to enable explainability of inferencing results, as it is desirable for expert systems to allow a natural language handling to challenge its results [6]. Therefore, as functions can be implemented in any way, it is their responsibility to explain their results by providing a natural language explanation of their inference result, which is then linked to the output individual.

So far, the focus of this section was on single functions and their execution in the knowledge base of ODD-ES. A function was given as an example which determines via a 'Fever Threshold' whether a 'Fever' is present. When using this to express expert knowledge, a large number of functions is required that build on each other. Thus, the 'Fever' could be used as an input of another function which concludes on the suspected diagnosis 'Covid-19'. This can lead to complex dependencies that need to be managed during maintenance. In order to improve the overview of such dependencies, they should be made explicit in the knowledge base. For this purpose, output individuals are linked to input individuals if they instantiate the same domain-specific class. This results in a network of functions that can be visualized for maintenance work and therefore could facilitate the identification of dependencies and an estimation of the effect of changes. Regarding the inferencing procedure, this structure opens up the possibility to perform an inferencing procedure based on value propagation. This is illustrated in figure 5, in which the inference engine initially writes the attribute values available in ODD-BP to corresponding input nodes in ODD-ES based on its previously established linkage. Afterwards, the inference engine executes the associated function 'Fever Threshold' and adds the output value to its output individual. This value is then propagated to the linked input individuals of other functions where this procedure is repeated. Thus, attribute values coming from ODD-BP are propagated through the knowledge base of ODD-ES while being modified by interconnected functions. If an already known attribute value changes in ODD-BP, only those functions that are involved in the propagating inference procedure need to be re-executed. This is an advantage compared to classical ontology-based inference, because there the entire knowledge base must be re-evaluated in the event of a single change. Another advantage of this propagating inference is its parallelizability. Using the example from figure 5, it would be possible to parallelize the functions for identifying 'Covid-19' and 'Febrile Seizure' as they are independent of each other.

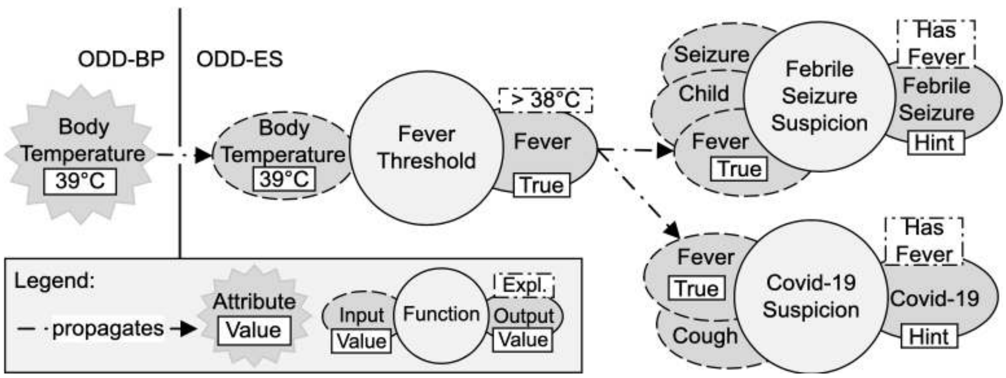

Figure 5: Inferences based on Value Propagation

As discussed, functions in ODD-ES can be used to describe symptom combinations that lead to the inference of suspected diagnoses. However, as soon as more than one affected person exists in an ODD-BP process instance, inferences can get incorrect. This is due to the assumption that any attribute value from ODD-BP can be linked to any input parameter in ODD-ES as long as they instantiate the same domain-specific class. Using the example of figure 5, it is possible that the function ‘Covid-19 Suspicion’ gets inserted the ‘Fever’ of one patient and the ‘Cough’ of another. To avoid this, a concept is needed to bind a set of functions to a type of dataobject. For this purpose ODD-ES uses blank nodes to link a set of functions to a domain-specific class of dataobjects. If a dataobject of this type occurs several times, the associated functions have to be copied accordingly. When establishing the links for inferencing, the inference engine only regards the attributes that belong to the dataobject for which the functions have been copied for.

5.3. Copilot

At the beginning of this chapter, ODD-ES was introduced with a focus on supporting the call taker in a decision-oriented creation of his/her mental picture. A central component of this is the Copilot which analyzes the knowledge base as a foundation for a consultation of the call taker about possible decisions and clarifications. The findings that the Copilot gets from this consultation are then used to modify the process instance. If this concerns an inferred decision, the value of the output individual, on which the inference has been materialized, is written to the corresponding attribute in the process instance by using their linkage. In case that a gap in the artificial mental picture was found that should be clarified by asking further questions, ODD-ES builds on an iterative procedure to modify the process instance that is described in the following.

Figure 6 depicts the function of a ‘Covid-19 Suspicion’ that has already been shown in figure 5 but this time with a focus on the ‘Cough’ of the patient. Since it is not yet known whether the patient has a ‘Cough’, the function cannot conclude a suspicion for ‘Covid-19’. However, since he or she has ‘Fever’, the function outputs that there is a ‘Hint’ for ‘Covid-19’. During the analysis of the knowledge base the Copilot identifies that this gap of the artificial mental picture could be clarified by further questions and proposes this as a new goal to the call

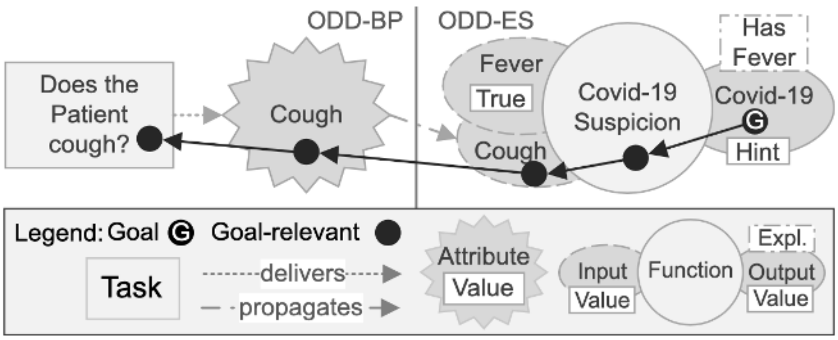

Figure 6: Modification of ODD-BP Process Instance

taker. If the call taker accepts this, the Copilot marks the output individual 'Covid-19' as a goal. Afterwards, the Copilot performs an iterative backward traversal through the network of functions in which all elements are marked as goal-relevant if they are unknown or, in case of tasks, unexecuted. A comparable mechanism is already implemented in ODD-BP and would only require slight adjustments to apply it to ODD-ES as well [4]. The established links between attributes and input and output parameters of functions thereby allow including the process instance directly in the iterative procedure. Further, since this solution is based on an already existing mechanism in ODD-BP, these modifications introduced by ODD-ES are taken into account when recommending next questions. Using the example of figure 6, the question about the patient's 'Cough' would become goal-relevant and recommended by ODD-BP, as it provides an attribute that could lead to the inference of 'Covid-19' in ODD-ES.

6. Future Work

So far, ODD-ES has a strong focus on inferences that address either a single or all dataobjects in a process instance depending on whether an inference function is linked to a blank node or not. In order to verify that this expressiveness is sufficient to support medical emergency calls, ODD-ES must be extensively used to express required expert knowledge. Further focus of our research will address the Copilot and especially its role in the context of an integration of symbolic and subsymbolic AI in the knowledge base. In this context, a 'hint' could for example also be inferred if a neural network makes a diagnosis, but then cannot explain it sufficiently, so that the call taker rejects it. This 'hint' could then be followed up by means of symbolic AI in order to underpin the suspected diagnosis with understandable facts.

7. Conclusion

In this paper, we introduced ODD-ES – an expert system that extends ODD-BP and aims to support call takers in medical emergency calls with adaptive questionnaires and recommended decisions that are derived from inferred artificial mental pictures. While artificial mental pictures result from an application of formalized expert knowledge to emergency-relevant information, possible questions and decisions are derived and discussed with the call taker through a component called Copilot. ODD-ES is in that sense ontology-driven as it uses domain-specific ontology classes to integrate the elements in its knowledge base with each other and with process instances in ODD-BP. ODD-ES is data-driven as it allows an inference procedure based on value propagation between semantically modeled functions that integrate symbolic and subsymbolic AI to support medical emergency calls.

Acknowledgments

This work is funded by the Federal Ministry for Economic Affairs and Climate Action under grant No. 22973 SPELL.

References

[1] T. P. Møller, H. G. Jensen, S. Viereck, F. Lippert, D. Østergaaard, Medical dispatchers' perception of the interaction with the caller during emergency calls - a qualitative study, Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine 29 (2021) 1–10.

[2] A. Baumann, Leitlinien oder Protokolle zur Notrufabfrage, in: A. Hackstein, H. Sudowe (Eds.), Handbuch Leitstelle, volume 2, 2017, pp. 189–193.

[3] B. Mayr, Strukturierte bzw. standardisierte Notrufabfrage: Leisten die Systeme tatsächlich, was sie vorgeben zu leisten?, Notfall und Rettungsmedizin 23 (2020) 505–512.

[4] E. Rietzke, C. Maletzki, R. Bergmann, N. Kuhn, Execution of Knowledge-Intensive Processes by Utilizing Ontology-Based Reasoning: ODD-BP: An Ontology- and Data-Driven Business Process Model, Journal on Data Semantics 10 (2021) 3–18.

[5] E. Rietzke, Modellierung, Visualisierung und Ausführung wissensintensiver Prozesse unter Verwendung Semantischer Technologien, Doctoral thesis, Universität Trier, 2021.

[6] F. N. Ford, Decision support systems and expert systems: A comparison, Information and Management 8 (1985) 21–26.

[7] S. N. Blomberg, H. C. Christensen, F. Lippert, A. K. Ersbøll, C. Torp-Petersen, M. R. Sayre, P. J. Kudenchuk, F. Folke, Effect of Machine Learning on Dispatcher Recognition of Out-of-Hospital Cardiac Arrest during Calls to Emergency Medical Services: A Randomized Clinical Trial, JAMA Network Open 4 (2021).

[8] B. S. Zachariah, P. E. Pepe, The development of emergency medical dispatch in the USA: a historical perspective, European Journal of Emergency Medicine 2 (1995) 109–112.

[9] T. Luiz, H. Marung, G. Pollach, A. Hackstein, Implementierungsgrad der strukturierten Notrufabfrage in deutschen Leitstellen und Auswirkungen ihrer Einführung, Der Anaesthesist 68 (2019) 282–293.

[10] S. N. Blomberg, F. Folke, A. K. Ersbøll, H. C. Christensen, C. Torp-Pedersen, M. R. Sayre, C. R. Counts, F. K. Lippert, Machine learning as a supportive tool to recognize cardiac arrest in emergency calls, Resuscitation 138 (2019) 322–329.

[11] C. Di Ciccio, A. Marrella, A. Russo, Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches, Journal on Data Semantics 4 (2015) 29–57.

[12] R. Vaculín, R. Hull, T. Heath, C. Cochran, A. Nigam, P. Sukaviriya, Declarative business artifact centric modeling of decision and knowledge intensive business processes, in: IEEE 15th International Enterprise Distributed Object Computing Conference, 2011, pp. 151–160.

[13] V. Künzle, M. Reichert, PHILharmonicFlows: Towards a framework for object-aware process management, Journal of Software Maintenance and Evolution 23 (2011) 205–244.

[14] L. Chen, D. Lu, M. Zhu, M. Muzammal, O. W. Samuel, G. Huang, W. Li, H. Wu, OMDP: An ontology-based model for diagnosis and treatment of diabetes patients in remote healthcare systems, International Journal of Distributed Sensor Networks 15 (2019).